作业二

2215408108 软件工程 程乐怡

4.3-6

选择 n_1 使得当$n \geq n_1$时，$n^2+17 \leq 3n^4$ 接下来找到常数 c 和 d，使得

$$T(n) \leq cn\log n + d。$$

根据递推关系，有

$$T(n) = 2T(n^2+17) + n$$

将$T(n)$的定义代入得到：

$$T(n) \leq 2\left(c(n^2+17)\log(n^2+17) + d\right) + n$$

进一步展开，可以得到：

$$T(n) \leq 2c(n^2+17)\log(n^2+17) + n$$

对n^2+17 进行近似处理，可以得到：

$$T(n) \leq cn\log(n^2+17) + 17c\log(n^2+17) + n$$

进一步处理得到：

$$T(n) \leq cn\log(3n^4) + 17c\log(3n^4) + 2d + n$$
$$T(n) \leq cn\log n + d + cn\log(3^4) + 17c\log(3n^4) + n$$

取$c = 2\log(3^4)$ 和 $d = 34$ 得到：

$$T(n) \leq cn\log n + 17c\log n + n$$

由于$\log(n) = o(n)$，存在n_2使得当 $n \geq n_2$ 时,

$$17c\log n \leq n$$

令$n_0 = \max(n_1, n_2)$ 有当$n \geq n_0$ 时,

$$T(n) \le cn\log n + d$$

故$T(n) = O(n\log n)$得证

4.4-2

一个深度为 i 的节点的子问题大小为$\frac{n}{2^i}$。

因此，树有 $\log n + 1$ 层，并且深度为 $\log n = 1$ 的叶子节点有 1 个。

对于$i = 0, 1, 2, \ldots, \log n - 1$ 的所有深度 i 的节点，总成本为

$$\sum_{i=0}^{\log n - 1} \left(\frac{1}{4}\right)^i \left(\frac{n}{2^i}\right)^2 = \sum_{i=0}^{\log n - 1} \left(\frac{1}{4}\right)^i n^2 = \left(\frac{1}{4}\right)^i n^2$$

$$T(n) = \sum_{i=0}^{\log n - 1} \left(\frac{1}{4}\right)^i n^2 + \Theta(1) < \sum_{i=0}^{\infty} \left(\frac{1}{4}\right)^i n^2 + \Theta(1) = \frac{1}{1 - \left(\frac{1}{4}\right)} n^2 + \Theta(1) = \Theta(n^2)$$

猜测 $T(n) \le cn^2$。

由此,

$$\begin{aligned} T(n) &\le c\left(\frac{n}{2}\right)^2 + n^2 \\ &= \frac{cn^2}{4} + n^2 \\ &= \left(\frac{c}{4} + 1\right) n^2 \le cn^2 \end{aligned}$$

最后一步在 $c \ge \frac{4}{3}$ 时成立。

4.4-3

一个深度为i 的节点的子问题大小为 $\frac{n}{2^i}$。

因此，树有$\log n + 1$层，并且叶子节点的数量为 $4^{\log n} = n^2$。

对于 $i = 0, 1, 2, \ldots, \log n - 1$的所有深度$i$ 的节点，总成本为

$$4^i\left(\frac{n}{2^i} + 2\right) = 2^i n + 2 \cdot 4^i$$

$$
\begin{aligned}
T(n) &= \sum_{i=0}^{\log n-1} \left(2^i n + 2\cdot 4^i\right) + \Theta(n^2) \\
&= \sum_{i=0}^{\log n-1} 2^i n + \sum_{i=0}^{\log n-1} 2\cdot 4^i + \Theta(n^2)
\end{aligned}
$$

这可以简化为：

$$
\begin{aligned}
T(n) &= (2^{\log n}-1)n + 2\cdot\frac{4^{\log n}-1}{4-1} + \Theta(n^2) \\
&= (2^{\log n}-1)n + \frac{2}{3}(4^{\log n}-1) + \Theta(n^2) \\
T(n) &= (n-1)n + \frac{2}{3}(n^2-1) + \Theta(n^2) = \Theta(n^2)。
\end{aligned}
$$

猜测 $T(n) \le c(n^2-dn)$。

$$
\begin{aligned}
T(n) = 4T\left(\frac{n}{2}+2\right) + n &\le 4c\left[\left(\frac{n}{2}+2\right)^2 - d\left(\frac{n}{2}+2\right)\right] + n \\
&= 4c\left(\frac{n^2}{4} + 2n + 4 - \frac{dn}{2} - 2d\right) + n \\
&= cn^2 + 8cn + 16c - 2cdn - 8cd + n \\
&= cn^2 - cdn + 8cn + 16c - cdn - 8cd + n \\
&= c(n^2-dn) - (cd-8c-1)n - (d-2)\cdot 8c \\
&\le c(n^2-dn)
\end{aligned}
$$

当且仅当$cd-8c-1\ge 0$ 时成立。

4.4-4

深度为 i 的节点的子问题大小为 $n-i$。

因此，树有 $n+1$层（$i=0,1,2,\ldots,n$），并且有 2^n个叶子节点。

对于 $i=0,1,2,\ldots,n-1$的所有深度i 的节点，总成本为 2^i 。

第n 层有2^n个叶子节点，每个节点的成本为 $\Theta(1)$，因此第 n 层的总成本为$\Theta(2^n)$ 。

将递归树的所有层的成本相加，得到：

$$
\begin{aligned}
T(n) &= \sum_{i=0}^{n-1} 2^i + \Theta(2^n) \\
&= \frac{2^n - 1}{2 - 1} + \Theta(2^n) \\
&= 2^n - 1 + \Theta(2^n) \\
&= \Theta(2^n)
\end{aligned}
$$

猜测 $T(n) \le c \cdot 2^n - d$。

$$
\begin{aligned}
T(n) &\le 2(c \cdot 2^{n-1} - d) + 1 \\
&= c \cdot 2^n - 2d + 1 \\
&\le c \cdot 2^n - d
\end{aligned}
$$

最后一步在 $d \ge 1$ 时成立。故 $T(n) = O(2^n)$ 得证。

4.4-5

通过递归树，非平衡下降，既不是满二叉树也不是多项式。可以证明 $O(2^n)$和 $\Omega(n^2)$。

为了证明这个上界非常紧致，可以证明没有其他选择。如果假设 $T(n) \le c \cdot n^k$，当将其代入递推关系时，新生成的 n^k系数的最大值为 $c \cdot (1 + \frac{1}{2^k})$，无论如何选择 c，这个值都大于 c。

猜测 $T(n) \le c \cdot 2^n - 4n$：

$$
\begin{aligned}
T(n) &\le c \cdot 2^{n-1} - 4(n-1) + c \cdot 2^{n/2} - 4 \cdot \frac{n}{2} + n \\
&= c \cdot (2^{n-1} + 2^{n/2}) - 5n + 4 \\
&\le c \cdot (2^{n-1} + 2^{n/2}) - 4n (n \ge 1/4) \\
&= c \cdot (2^{n-1} + 2^{n-1}) - 4n \\
&\le c \cdot 2^n - 4n \\
&= O(2^n) (n \ge 2)
\end{aligned}
$$

猜测$T(n) \ge c \cdot n^2$：

$$
\begin{aligned}
T(n) &\ge c \cdot (n-1)^2 + c \cdot \left(\frac{n}{2}\right)^2 + n \\
&= c \cdot n^2 - 2cn + c + \frac{c \cdot n^2}{4} + n \\
&= \frac{5}{4} c \cdot n^2 + (1 - 2c)n + c \\
&\ge c \cdot n^2 + (1 - 2c)n + c (c \le 1/2) \\
&\ge c \cdot n^2 = \Omega(n^2)
\end{aligned}
$$

4.5-1

1. $T(n) = 2T(n/4) + 1$

$$\Theta(n^{\log_4 2}) = \Theta(\sqrt{n})$$

2. $T(n) = 2T(n/4) + \sqrt{n}$

$$\Theta(n^{\log_4 2} \lg n) = \Theta(\sqrt{n} \lg n)$$

3. $T(n) = 2T(n/4) + n$

$$f(n)\text{阶数高于}n^{\log_4 2}\text{，}T(n) = \Theta(n)$$

4. $T(n) = 2T(n/4) + n^2$

$$f(n)\text{阶数高于}n^{\log_4 2}\text{，}T(n) = \Theta(n^2)$$